

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
3 January 2002 (03.01.2002)

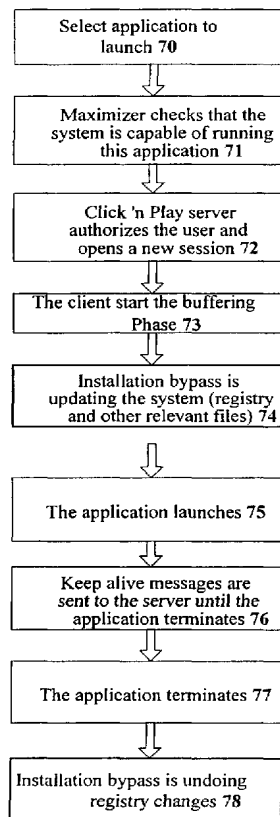
PCT

(10) International Publication Number
WO 02/01350 A1

- (51) International Patent Classification⁷: **G06F 9/00**
- (21) International Application Number: PCT/US01/20036
- (22) International Filing Date: 22 June 2001 (22.06.2001)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
60/213,989 26 June 2000 (26.06.2000) US
60/213,990 26 June 2000 (26.06.2000) US
- (71) Applicant (for all designated States except US): **GAME-PLAY TECHNOLOGIES LTD.** [IL/IL]; 21 Habarzel St., 69710 Tel Aviv (IL).
- (71) Applicant (for TJ only): **FRIEDMAN, Mark, M.** [US/IL]; 1 Alharizi St., 43406 Raanana (IL).
- (72) Inventor; and
(75) Inventor/Applicant (for US only): **ANTEBI, Liran** [IL/IL]; 29 Yom Hakipurim St., 58485 Holon (IL).
- (74) Common Representative: **FRIEDMAN, Mark, M.**; c/o Castorina, Anthony, 2001 Jefferson Davis Highway, Suite 207, Arlington, VA 22202 (US).
- (81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.
- (84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE,

[Continued on next page]

(54) Title: A SYSTEM AND METHOD FOR ENABLING RAPID LAUNCHING AND EXECUTION OF STREAMED APPLICATIONS-ON-DEMAND



(57) Abstract: A system and method for enabling rapid launching and execution of applications-on-demand over a network. The service of the present invention allows PC users with a broadband Internet connection to rapidly launch and play PC games etc. (70) without the need to leave their homes to buy the games. Based upon a proprietary Installation Bypass Mechanism (IBPM) (74) and a Game Traffic Analyzer (GDTA) module, the Click'n Play system streams games effectively to the user's computer from a remote server over any type of broadband network. The modules analyze the client computer setup (71) and install critical files on the client computer, such as drivers, game files, and updating the registry (74), in order to enable fast start up and secure execution of interactive broadband content. The modules enable the extraction of game files after completion of a game (78), such that the client computer registry and memory are returned to their prior condition, and data flow is optimization over a network.

WO 02/01350 A1



IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Published:

- *with international search report*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

A SYSTEM AND METHOD FOR ENABLING RAPID LAUNCHING AND EXECUTION OF STREAMED APPLICATIONS-ON-DEMAND

FIELD AND BACKGROUND OF THE INVENTION

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a system for optimizing the streaming of games and other media-rich applications to a user's computer from a remote server over any type of broadband network.

2. Description of the Related Art

Application streaming services have become increasingly popular with the progress of server-client based applications on IP networks. Historic problems of low bandwidth and high traffic have prevented mass adoption of application streaming services for multimedia applications. Nevertheless it is assumed that broadband networks will continue to penetrate the market, enabling ever improved data transfer possibilities.

One of the most popular uses for domestic broadband usage is the access to multimedia resources on the Internet. Music files, for example, can now be easily transferred across broadband networks. However, video and interactive game files cannot, even in a typical broadband environment, be effectively transferred to users.

With the impending adoption of broadband networks, many attempts have been made to provide multimedia-streaming applications for such networks. There are currently several known companies that offer technology or services in this field, including:

Media Station (www.mediastation.com) offers an application streaming service on broadband networks. Their service is called SelectPlay and is available to users throughout the U.S. The service is offered via a web site and offers a variety of games and other applications. However, SelectPlay does not include an installation bypass mechanism and therefore the user needs to install the application in order to use it. This installation prohibits immediate/rapid launching of games. When a SelectPlay user wants to use an application that is offered to him or her by this service through the web site (the

first time s/he selects this application), s/he receives an installation screen that is identical to the game installation in the boxed version. The user needs to go through the entire installation process, as if s/he was installing the application from a local CD-drive. Only when the installation is complete, the user can run the application and stream it from the SelectPlay servers.

Into Networks (<http://www.intonetworks.com/>) offers a game streaming service for broadband users, called PlayNow. The system, IntoMedia, is also offered to service operators for licensing. There is no indication that the IntoMedia system has a mechanism that determines which application data is streamed before the application is executed, and which application data is streamed while the application is executed, according to data flow patterns of the specific application, resulting in inefficient streaming of some applications, such as games.

Stream Theory (www.streamtheory.com), use a technology is called StreamIt, used for streaming of multimedia content via broadband networks. This system, however:

- Does not detect the absence or presence of DirectX (which is a set of multimedia programming interfaces from Microsoft for Windows 95/98, NT and 2000 that provide low-level access to the hardware for improved performance) and therefore does not offer DirectX for download.
- Requires computer rebooting after driver installation, which is done once before the user can activate the system.
- Does not enable CD-ROM emulation, which means the system cannot play audio tracks, and therefore cannot provide a complete application usage experience that is identical to using a boxed application. According to this invention, audio tracks are stored in a different format than the rest of the application data and therefore cannot be read by a standard driver. Only the use of a special CD emulation driver enables reading these files (and playing CD tracks as a result).

Real Networks (www.real.com) develops and markets software products and services designed to enable PC users to send and receive multimedia services using the Web. Their system, RealArcade, is different than the other systems in the sense that the games must be built specifically for this service and the service is provided only through

RealNetworks network. Also, based on the information that is currently available, the RealArcade does not seem to stream games while they run, but rather download them in a more traditional manner. The RealArcade system requires significant changes in the streamed application, mainly for interfacing with their system from within the application. Therefore, in order to have an application streamed through RealArcade, it has to be heavily modified, to the extent that modifying existing applications is not practical. Real Networks are therefore offering developers to develop new applications using a special SDK, and are not targeting the market of existing titles.

There is thus a widely recognized need for, and it would be highly advantageous to have, a system that can enable a means for efficiently enabling the execution of existing games on a PC, where the games are streamed from a server computer, such that a fast or close to immediate start-up time for launching games played on PC's is enabled, and where games are played with high stability. Furthermore, there is a need to achieve this without over-burdening or otherwise negatively affecting the client computer. There is also a widely recognized need for an intelligent way of streaming game flow to a PC from a server, wherein critical files are executed on the client computer, and remaining files are executed on the server.

SUMMARY OF THE INVENTION

According to the present invention there is provided a system (hereinafter referred to as "Click 'n Play") and method for enabling rapid launching and execution of streamed applications-on-demand over a network. The service of the present invention allows PC users with a broadband Internet connection to rapidly launch and play PC games without the need to ever leave their homes to buy the games. Based upon a proprietary Installation Bypass Mechanism (IBPM) and a Game Data Traffic Analyzer (GDTA) module, the Click 'n Play system streams games effectively, almost immediately after launching, to the user's computer from a remote server over any type of broadband network.

The present invention avoids the downloading phase of many of the prior art technologies, enabling the user to go directly to 'stage 2', i.e. immediately executing the application without the need to install it first. Furthermore, the installation of the Click 'n

Play driver of the present invention is designed in such a way that it provides the same or additional functionality as compared to prior art technologies, but does not require rebooting of the computer after the installation. In addition, the present invention enables the usage of existing applications, without the need for modifying these applications for usage with the Click 'n Play system.

According to the present invention, a method and system is provided for enabling the streaming of games to a user's computer, such that both fast start-up time and high stability are achieved, and such that the user/client computer is not over-burdened or otherwise negatively affected by the game. The user or client computer, according to the present invention, includes any multimedia-compatible computing device, such as a PC, Internet TV, notebook computer, personal handheld assistant, smart phone, wearable computer, and mobile computer. The present invention enables such a method and system by providing an Installation Bypass Mechanism (IBPM), which is a software component for analyzing the client computer setup, and accordingly installing critical files on the client computer, such as drivers, game files, and updating the registry, in order to enable fast start up and secure execution of interactive broadband content. Moreover, the IMPM enables the extraction of game files after completion of a game, such that the client computer registry and memory are returned to their prior condition.

An additional module of the present invention is the Game Data Traffic Analyzer (hereinafter the "GDTA"), which is a back office tool that analyzes game flow traffic by automatically tracking game access to the resources found on the local media. This module effectively manages the data streaming and storage processes, so that game flow data is optimized and high quality secure data transfer is enabled. The GDTA module decides which files should be part of the initial streaming, and which files will be streamed during game playing. It also decides which files remain on the user's machine (in a cache) and which will remain on the server to be streamed when required.

From the user's point of view, the use of Click 'n Play is as follows: It begins with the installation of the Click 'n Play client. The user downloads a small self-extracting file from a website and follows a simple installation process. The Click 'n Play driver is installed in a designated Windows library while the rest of the files (e.g. the game cache)

can be located according to user preferences. Installing the client does not interfere with any other hardware/software components. Once this is completed, the user can start using the system - the user selects a game from a web-based list, the game begins streaming from the server and shortly after, the game starts. The game automatically starts once the essential game files have been buffered on the user's PC. Only a small fraction of the game's files are required for the game to start – on average, about 30MB. The rest of the files reside on the Click 'n Play server, which functions as a 'virtual drive' for this part of the game data.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is herein described, by way of example only, with reference to the accompanying drawings, wherein:

FIGURE 1 illustrates the server structure according to the present invention.

FIGURE 2 illustrates the client structure according to the present invention.

FIGURE 3 illustrates the system software components, including the IBPM and GDTA modules, according to the present invention.

FIGURE 4 illustrates the flow of the content aggregation process, in which the tracking process is done, according to the IBPM and GDTA modules.

FIGURE 5 illustrates a usage flow of the Click 'n Play system.

FIGURE 6 describes the Analyzing Process, according to the IBPM.

FIGURE 7 illustrates the flow of the game execution process, including the installation bypass actions.

FIGURE 8 illustrates a general flow diagram of the analyzing process, according to the GDTA.

FIGURE 9 describes the flow for creating the list, according to the GDTA of the present invention.

FIGURE 10 describes an example of the generation of a file access pattern.

DESCRIPTION OF THE PREFERRED EMBODIMENT

The present invention relates to a system and method for enabling rapid launching and execution of streamed applications-on-demand over a network.

The following description is presented to enable one of ordinary skill in the art to make and use the invention as provided in the context of a particular application and its requirements. Various modifications to the preferred embodiment will be apparent to those with skill in the art, and the general principles defined herein may be applied to other embodiments. Therefore, the present invention is not intended to be limited to the particular embodiments shown and described, but is to be accorded the widest scope consistent with the principles and novel features herein disclosed.

Specifically, the present invention can be used to enable the streaming of applications, including media rich applications (such as games, educational software, etc.) from a remote server (over any type of broadband network) to a user's computer. According to the present invention, a method and system is provided for enabling this streaming of games to a user's computer, such that both fast start-up time and high stability are achieved, and such that the client computer is not over-burdened or otherwise negatively affected by the game.

The principles and operation of a system and a method according to the present invention may be better understood with reference to the drawings and the accompanying description, it being understood that these drawings are given for illustrative purposes only and are not meant to be limiting, wherein:

Figure 1 illustrates the Click 'n Play streaming server components of the Click 'n Play system, wherein:

- **Database 10** – contains user information, system information, IBPM data and usage statistics.
- **DBAL 11** – (Database Abstraction Layer) an application program interface (API) that allows applications to access the database10 without being depended on the database specific database implementation and structure. This interface is designed by the inventors and is specific to the system of the present invention, however it is not innovative, because the interface design, once the system functionality and requirements are defined, is straightforward.

- **Virtual File System 12**– a driver that communicates with the Click 'n Play driver on the client side in order to transfer data that the streamed application requires. The driver is designed by the inventors since it works with the proprietary file system, but it is relatively straight forward to design.
- **Server Application 13**– the application that is responsible for managing the streaming of all the applications that are requested by users.
- **Web Server 14**– running the site that is the Click 'n Play front end – the place where the user logs in, selects and executes the application.

Figure 2 illustrates the components of the Click 'n Play client system, wherein:

- **Application Level 20**– includes the client side of the Maximizer **21** (the Maximizer is a unique component that enables added functionality. However, implementation of the Maximizer, both on the server and client sides, is achieved using existing knowledge that is already implemented by us and others in stand-alone offline applications, such as boxed games), which analyzes the user's computer; the Web Interface ActiveX component **22**, which is responsible for running the Click 'n Play driver once the user selected an application to stream, and the application **23**, which is the executable file of the application the user selected to stream. The inventors designed this Interface ActiveX component, which it is a software component that is rather obvious to create, and is used as a tool for running the driver on the client machine.
- **Driver Level 24**– includes the security components **25** which protect the system from illegal use or hacking), the Compression Engine **26** (which decompresses the data that was streamed from the server) and the File System Driver **27** (which sends the data to the application). This driver on the client side provides unique functionality while integrating tightly within a Windows operating system environment.
- **System Resources Level**– includes Windows services for storing and retrieving data from the local hard drive **28** and for sending and receiving data from the Internet, via the Windows Network Layer **29**.

As can be seen in **Figure 3**, the present invention is comprised of three levels of software components: the Back Office utility suite **30**, the click 'n Play system components **31**, and external components **32**. As can be seen, there are two vital modules in the utility suite **30**, Installation Bypass Mechanism (hereinafter the "IBPM") **33**, which is a software component for enabling application preparation and execution, and a Game Data Traffic Analyzer (hereinafter the "GDTA") **34**, which is a back office tool that encodes applications for optimal streaming. The back-office utility **30** uses a device driver **36** that hooks to a windows file system in order to receive every file system request. The device driver **36**, which is the heart of the system, is used by the analyzer to track game access to files. The driver **36** is a file system hook. The hook receives all the file system based requests from Windows applications. The requests types are Open, Read and Write. In order to analyze the game resources access pattern for a specific game, we specify for the analyzer in which folder we installed the game (for instance C:\Program Files\Jane's Combat Simulations\USAF). The device driver **36** filters all windows files system requests and records only the requests that are made to the specified folder. When the game is executed (for porting purposes), the analyzer records all the requests that were made by the game.

The driver **36** records only read and write requests. The following data is logged for every request:

- 1) Request time
- 2) Request type (Read/Write)
- 3) File name
- 4) Start offset
- 5) Length

The information is stored in a database, such as MS Access database, that is used only for logging and processing this information. It is part of the back office utility suite **30**, and processed by the GDTA **34** when the game terminates.

The back-office tools (utility suite **30**) thereby executes the back-end functions of preparing applications for streaming, according to the method of the present invention. This preparation requires that the IBPM **33** links and incorporates external components,

such as a snapshot utility 37 for capturing the changes that the game installation process makes to the system. These changes include changes to registry and other files (such as .INI files), in order to map the changes done by the original installation in such a way that could be imitated by the installation bypass process.

The respective functions of the two primary modules, the IBPM 33 and GDTA 34 can be seen with reference to **figure 4**, wherein is illustrated the flow of the content aggregation process, in which an application is encoded, or prepared for running on the Click 'n Play system. As can be seen in figure 4, the product installation 41 is followed by both product analyzing 42, which determines which files should be streamed to the application before it starts (priority data), and registry tracking 43, which compares the registry fields and their values before and after the product installation, in order to identify the registry changes that the product makes during installation. The product installation stage 41 is the process whereby files are prepared for installation, and is executed by the IBPM, as is described below. The product analyzing stage 42, wherein data flow is analyzed and optimized for execution, is executed by the GDTA, as described below. Following the product analyzing 42 stage, the IBPM initiates product compression 43, which significantly decreases the size of product files by compressing each one of them (in order to decrease the amount of data that is streamed from the server to the client), and CD-image(s) creation 44, which is a process where an emulated image of the original product CD is created on the server. This is required in order for the product to run on the client machine although all its data is stored on the server.

Accordingly, **figure 5** illustrates the Click 'n Play system, as seen from the user's point of view. It begins with the installation of the Click 'n Play client. The user downloads a small (for example 1.25MB) self-extracting file from a website (or from a CD ROM) and follows a simple installation process. The Click 'n Play driver is installed in a designated Windows library while the rest of the files (e.g. the game cache) can be located according to user preferences. Installing the client *does not* interfere with any other hardware/software components. Once this is completed, the user can start using the system, as follows:

The user initially selects a game from a server based online list **501**. In order to aid the user's decision making, the user may choose to view the compatibility of a chosen game to his/her system **502**, as well as estimated download **503** time of a chosen game. Once the game has been selected **504**, the server buffers essential files **505**, such that a minimal yet effective selection of files will be downloaded to the users computer. In this way a fast start-up for the game is insured. Once the buffering is completed **506**, the game can begin running **507**. During play, various parameters are monitored and queried, according to the personal profile of the player, such as user subscription **508**, client computer disk space **509**, hacker activity **510**, and video data messages **511**. Actions are executed according to results of the monitoring or queries. In the event that subscription has ended **515**, disk space is too limited **516** or hacker activity is detected **517**, the user is disconnected **518** from the Click 'n Play Streaming Server. After play, when the user eventually decides to end the game **512**, at which time game summary information is displayed **513**. In the case where a video should be played **511** a video play message **520** is provided, followed by either playing the video **521** or returning to the game **522**.

Installation Bypass Mechanism IBPM

The main processes according to the IBPM are the installation tracking (encoding) and installation bypass (executing) processes, as described in detail below.

1. Installation Tracking

General

Installation tracking is a process that takes place each time a game is ported to the Click 'n Play system. The process is part manual and part automatic and it's purpose is to identify all the changes that the original application installation is doing during the installation process, as well as detecting other components it installs, in addition to the application itself (such as DLLs, drivers, etc.).

Manual Tracking

Manual tracking includes the following actions:

1. The person who performs the product preparation process (the Encoder) locates the minimum system requirements for the specific game, as listed by the game developer on the game box and/or inside the user manual, and feeds them to the

IBPM database. IBPM uses a proprietary automatic HW/SW diagnostic mechanism to make sure that the game can run on the user's computer. This mechanism queries Windows and device drivers such as audio and video drivers, in order to get the computer characteristics, such as RAM size, CPU speed, type of video card, type of sound card, version of DirectX drivers, etc.

2. Track software installations during the original installation. This is a backup procedure to the automatic detection process.

The results of this manual tracking are updated in the IBMP database and are used each time the game is executed via Click 'n Play on the user's machine.

Automatic Tracking

The automatic process tracks changes that the original game installation is doing to the system, mostly to the registry, in order to reproduce these changes on the user's system, when the user initiates a Click 'n Play game.

The automatic tracking is based on a Click 'n Play proprietary device driver that was created for the purpose of hooking the Windows registry services. The device driver records every registry change, such as creating keys/values, changing keys/values, etc.

The driver is initiated first, and following this, the game installation application is launched. The driver records every registry change that occurs at that time. Obviously, no other application is executed at that time, to isolate the changes that the game installation is doing.

As can be seen with reference to **figure 6**, the installation analyzer is started **61** before the installation of a game or application. Subsequently the original game installation is run **62**. During this original installation, the encoder (person managing the encoding process) logs additional applications **63** (manually). In order to achieve this, the IBPM operator follows the installation process and notes any installations that are additional to the game installation itself (drivers and applications), and if these installations are necessary or optional. Installation tracking is also done automatically and this process backs up the automatic tracking. The manual tracking is especially important in cases where the automatic tracking fails to detect an installation, for example, if the

installation does not update the registry. Following the logging of step 63 the game installation finishes 64, after which time the installation analyzer is terminated 65. System changes and minimum requirements are subsequently stored 66 in the IBPM section of the database. This final stage includes two parts: One is going over the results of the automatic tracking and verifying that it indeed recognized all installations. If not, the details of the installations that were not recognized are automatically added. The second part is adding the minimum configuration requirements to the IBPM database.

The Click 'n Play system, using the Installation Bypass Mechanism (IBPM) of the present invention, does not install the game in the traditional manner. Instead, it installs only a minimal amount of data initially on the client device, and the rest is streamed during game execution. Therefore a regular game installation is not necessary when using a game via Click 'n Play. However, a typical game installation handles more than just copying game resources to the hard drive - it analyzes the user's computer to determine if it can run the game, installs additional drivers, updates the registry and performs additional tasks. The installation bypass mechanism is the software component that handles these additional tasks.

The above mentioned tasks are achieved via a manual process as follows:

- 1) Installing the title on a "virgin" (clean with no software, other than what's required for this process to run) test computer, while using a 3rd party tool (such as the shareware version of Arkosoft's product called "System Snapshot" (www.arkosoft.com)). This tool captures the changes that the game installation process makes to the system, including changes to registry and other files (such as .INI files). At the end of this process the encoder gets from the Snapshot a list of files that were changed/added and a list of the registry keys that were added/modified.
- 2) The encoder (the person who initiates the encoding process) analyzes the file (manually), removing all the changes that were logged, but not done by the game installation (for example changes the operating system did at the same time). The result is a 'clean' list of registry keys and files that should be added/modified in order for the game to run properly.

- 3) This list is then added to an installation "script" file, which is used by Click 'n Play for each title. The installation bypass script file is an INI file (INItialization file- A file that contains startup information required to launch a program or operating system) that contains all the information that the installation bypass mechanism on the client side needs in order to perform the installation bypass before the application executes. These parameters are:
- Relative path to the application's executable (including command line parameters, if any exist in the original application).
 - Registry changes and additions – which keys were added/changed and what are their values.
 - Changes in game INI files that are done by the original application installation (parameters that the original application determines during the installation).
 - A list of changes that the original application installation is doing to the Win.ini and System.ini files.
 - A list of files that the original application installation is copying from the application CD to Windows folders – DLLs, INI files, etc.
- 4) The encoder now performs the Click 'n Play automatic installation process on another "virgin" machine, in order to reveal any incorrect registry change that might occur due to incorrect analysis. If the two machines are identical then the process is over. If not, the encoder fixes the changes list by adding removing the specific field that caused the difference.

2. Installation Tracking

During the Game Execution Process, the IBPM module executes the installation bypass script file, changing the registry and other system files. The IBPM thereby analyzes the client computer setup, and accordingly installs critical files on the client computer, such as drivers, game files, and updates to the registry. Moreover, the IMPM enables the extraction of game files after completion of a game, such that the client computer registry and memory drives are returned to their prior condition.

The flow of the game execution process, including the installation bypass actions, according to the present invention, can be seen with reference to **figure 7**. As can be seen in the figure, the user initially selects the application to launch **70**. Following this the maximizer **71** checks that the system is capable of running the application **71**. Based on the results of this check, the click 'n play server authorizes the user and opens a new session **72**. The client device launches the buffering phase **73** for the files that were previously downloaded into the client. While steps **70-73** have been in process, the IBPM updates the system, including the registry and other relevant files **74**. Once these previous steps are in place, the application launches **75** and the streaming of the files that are not found on the client continues from the server. During the streaming, keep alive messages are sent to the server until the application terminates **76**. These messages are simple small messages that do not contain any data. Their only purpose is to force the client to work only with the server. If for any reason these messages are not sent by the client (for example, if the network connection was lost) and confirmed by the server, periodically, the client application will terminate. This is a security mechanism that prevents the user from running the Click 'n Play offline and avoiding usage charges. Upon termination of the application **77**, the IBPM undoes the registry changes **78**, so as to leave the client device unaffected by the game.

The IBPM is responsible for performing all the installation tasks of a Click 'n Play game, which are otherwise performed by the original game installation. This does not include the installation of the game data itself, since Click 'n Play streams this data during game execution.

The IBMP is also responsible for retrieving game files from the client following application termination, so as to leave the user's computer to the same state it was before the 'bypass installation'. This is necessary because the idea behind the system is that the user's computer is affected by the game ONLY when the game is running. Once the game is completed, the computer must be automatically restored to its original state. This process is also done by the IBPM.

The above process takes place each time a game is executed via Click 'n Play. In addition, a preparation process is also required in order for IBPM 'installation bypass' to

work properly. This process monitors the original game installation process and determines which additional installations need to be done and which files need to be updated (mostly registry changes). This is a part of the IBPM that is activated at the porting process of a game to Click 'n Play environment. The preparation process is based on an automatic System Tracker that tracks the installation process, and on a manual update. The System Tracker is a Click 'n Play utility that tracks registry changes and files changes. The manual process is executed by the person who prepares the product for Click 'n Play (the Encoder) and essentially includes updating of the system requirements for running the game.

The tracked changes that are made by the original game installation are used to restore or recreate the changes on the user's computer, for the duration of the game. The changes are made before the game starts running, and are undone when the game ends. This allows the bypassing of the original installation, and the creating of a mirror installation process, which is both automatic and fast. The process is automatic because all the selections that the user does during a regular installation (selection of destination folder, graphic card setting, etc.) are already done in the product preparation stage, so that the user does not need to make any selections. This, in addition to the fact that, as opposed to a regular installation, files are not copied from the product CD to the hard drive, makes the installation process very fast (typically a few seconds).

The above-described method enables the offering of games in their original version, without the need to modify any of the game code, the installation code or the game resources. It eliminates the need for game developer involvement in the porting process, and makes the process itself simpler.

In order to affect the correct execution of such a bypass process, the present invention depends on 3 elements:

1. Accurate and complete detection of all the actions that the original game installation is performing.
2. Automatic execution of these actions by the IBPM in such a way that does not require user involvement.
3. Restoration of the system to its original state when the game terminates.

With these three elements, the IBPM of the present invention enables game flow improvements for all types of boxed games.

The IBMP therefore enables remote streaming of applications, while maintaining the original functionality of the product. The IBMP, however, relies on an additional module in order to be fully optimized and stable. Therefore the IBPM requires the input and support of a Game Data Traffic Analyzer (GDTA) module in order to enable the high quality secure and optimized game flow from remote servers. The actions that are performed on the game for the installation bypass preparation process (such as installing and running it) are also required for the GDTA, and therefore in the game preparation process, the GDTA is activated as part of the IBPM.

The Game Data Traffic Analyzer (GDTA)

The GDTA **34** is part of the set of back-office tools (utility suite) **30** that is used for porting a standard boxed game or application to the Click 'n Play system. This porting process mainly includes the creation of the access profile lists and the compression of game data. This tool needs to be executed for each game, and each time the network definitions change (for example, if the game is already running on a cable network and needs to be adapted to satellite network). The GDTA **34** encodes (executes the process of preparing an application for the Click 'n Play system, by analyzing applications) game flow traffic by automatically tracking game access to the resources found on the local media. This module effectively manages the data streaming and storage processes, so that game data flow is optimized and high quality secure data transfer is enabled. The GDTA **34** module optimizes the data streaming, by deciding which files should be part of the buffering stage, and which files will be streamed during game playing. The GDTA **34** module is not a necessary component for the click 'n Play system to function, however, it is such an important optimization that without it, games would not effectively run on the Click 'n Play system.

More specifically, the GDTA module **34**, during the Game Preparation Process, detects the traffic of data blocks from the hard drive and the CD-drive to the game application during game execution; and prepares a 'priority list' of which data blocks

should be buffered and which should be streamed during game execution. During the Game Execution Process, the GDTA module 34 does not function.

From the process point of view, these two components, IBPM 33 and GDTA 34, are steps in the game preparation (encoding) process for the Click 'n Play system. Both IBPM 33 and GDTA 34 are therefore parts of the encoding process, and comprise the encoding back office utility. In addition, the IBPM 33 module is responsible for the compression and streaming stages.

As can be seen in **Figure 3**, the GDTA 34 connects directly to the Click 'n Play system components, such as the device drivers 36 in order to track the flow of data that the game requires in order to determine which data needs to be streamed before the application can start running.

The GDTA component 34 is used for optimizing the flow of game data from the server to the user when a Click 'n Play game is ported, by:

- Tracking access to game resources using the Device Driver.
- Creating data access profiles for each game resource file.

Such optimization is required because of two contradicting requirements: On one hand, installing as much of the game data files before the game should be executed in order to provide delay-free execution of the game. However, this downloading results in an increased response time from when the user selects the game until the game actually starts to run. Even in broadband networks, this process is long when installing an entire game, and therefore requires a significant wait for the user until execution of the game is possible. On the other hand, if all game files are streamed during game execution, the game will instantly run, but will suffer occasional delays and sometimes even crash when required data blocks does not arrive on time.

The GDTA was created to decide which files should be part of the initial installation and which are to be streamed. It also decides which files will remain on the user's machine (be cached) and which will need to be streamed each time they are required by the game.

The GDTA implements the idea of optimizing network traffic by automatically tracking game access to the resources found on the local media. If the access pattern of

the game application to its resources is known in advance, the network access can be optimized, and it can even be decided in advance which data the game needs. Currently this information is used for defining the data blocks that the application requires before it can start running. Using the appropriate algorithm, this information is also used for predictions, such as bringing data on-the-fly before the game requests it.

To identify the access pattern we need to analyze the game. This analysis is part of the content aggregation stage, which is required for preparing a game for the Click 'n Play system. Basically, the analyzing process means tracking of all hard drive accesses under the game installation directory made by the operating system and the product's executable(s) when the application is running. Output of this operation is a list of accessed files with specified offsets and sizes of blocks. This process is required for gathering information about priority data and the file access sequence. The GDTA 34 is generic and therefore can handle any type of game. However, to analyze the game it still requires a person to play the game while analyzing it.

According to the GDTA module, the analyzing process is as follows: the user needs to set the input and output directories names. Input directory is the directory where the product is installed and output is where the ANA-file will be stored. The ANA-file stores all the data about which files the product is reading during its execution. When the directories are set, the user may command an "Analyze" function (by clicking on an "Analyze" button), and following that, the Compression Tool initializes the QPFSH virtual device driver that tracks all accesses to the specified input directory. QPFSH is a file system hook driver that gets the information from the operating system about all accesses to the file system, and filters these accesses to a certain directory only. QPFSH is a Windows ring-0 level driver, and as such it receives all the requests to retrieve data from the hard drive. Each request refers to a specific part of a file that is located in a specific folder. Since QPFSH is a low level driver, it doesn't know which application requested data (and on the other hand, it must know what was requested by the product application only), but it does know from which folder the data was requested. We assume that each request that was made to the folder where the product is installed, is coming

from the product application. This is the reason that this directory is specified to the driver.

The Encoder runs the application, waiting for the main menu and presses a special hotkey to write the FAP separator in the ANA-file. The FAP separator is simply a sign inside the ANA-file that marks the point where the game finished reading all the files that were required in order to get to the main menu. After this, the Encoder can quit the product, if the priority data was defined as the data that the game requires until it gets to the main menu. Alternatively, the encoder can continue to use the data, until the point that was defined as such that all the data the game reads in order to get to it is priority data (for example the start of a game session, race, etc.), and then quits the product.

Preceding the present invention, no other application has been designed to automatically analyze the access patterns of games. The assumption (and confirmation in actual tests) that games have such patterns is a breakthrough. Secondly, in order to find a pattern (that can be useful for our target, as explained above), the right parameters for such analysis must be chosen. A lot of trial and error took place until we reached the correct tuning.

The Analyzing process, according to the GDTA

Figure 8 describes the General Flow Diagram of the analyzing process, according to the GDTA module. As can be seen in figure 8, the encoder initially specifies the game folder to the analyzer **81**, which is done by changing a parameter field in an ini file. Subsequently the analyzer is launched **82**, followed by the launch of a game **83**. The encoder then plays the game **84**, which entails playing a normal, but extensive sequence of the game. The game is subsequently terminated by the encoder **85**, after all game levels/arenas/missions are committed. The analyzer is subsequently stopped **86** and the results are processed **87**.

Processing Information from the Device Driver

When the game terminates, the analyzer gets a list of all the requests that were made by the game. Based on that list, the analyzer generates two additional lists:

- 1) Priority list, which includes resources that must be available before the game executes.

2) File access pattern.

The priority list is constructed in the following manner:

Each file is divided to 64Kb section (data block). The priority value is set to each data block separately, rather than the whole file. That way different data blocks of the same file can get different priority values and the distinction is more accurate.

The threshold priority value is set in advance. Another predefined parameter that can be used is the required initial loading time of the game. However, these two parameters are optional. By default, the priority list includes all data blocks.

Figure 9 describes the flow for creating the list. According to **figure 9**, the application requests the driver to provide a data block (a part of file). The driver retrieves this data block and then GDTA calculates and stores the block number for it **91**. GDTA also updates the total size of blocks retrieved so far **92**. After reading each data block GDTA determines if the application still requires data blocks, using a timeout mechanism (if a certain time passed without receiving any request for a new block then it assumes the application is not reading anymore data) **93**. If that's the case, GDTA stops the analyzing process **95**. If not, it continues by checking if all the priority blocks were already retrieved (either by reaching the maximum size limit defined in advance by the encoder for the total size of priority blocks, or by manual signal from the encoder – by pressing a key – when the application reaches the main menu, or a point that was defined as such that beyond it all the data that the application requires is not considered priority blocks) **94**. If one of these conditions apply, then the analyzing process terminates **95**, otherwise GDTA waits for the application to request another data block **91**.

For example, suppose the read request was for the file Usaf.exe, at offset 16384 and length 1024 (data block size is 4096). The block number, in this case, will be 5 (because 16384 (range 0 – 16383) is 4 blocks ($16384/4096=4$) so the next byte (#16384) is actually the beginning of block #5] and it will be a priority block. This block has priority since, in this example, an EXE file is always the application executable itself, and an application cannot start executing before its entire executable file is streamed. From when the application starts, each block that is requested by it is given a number which is its order in the request list. The above number (block 5) was given as an example.

The File access pattern is then generated as follows:

The access pattern for every file is logged. This is done by sorting the analyzer list by time (as primary key) and then by files (as secondary key). The result is list per file, which is actually the file's access pattern.

Figure 10 illustrates an example for the file usaf.exe where standard block size is 4096 bytes. The recorded actions can be seen in figure 10, such that the priority list generated in the usaf.exe the list will be 1,2,3,4,5,1,6.

The information gathered by the analyzer is used by the Click 'n Play system to efficiently download game resources with minimal network traffic.

The priority list is downloaded before the game starts and is used by the prediction mechanism to ensure smooth and delay free play. The list is also used by the server to prevent hacking attempts, by tracking the access pattern of the game and comparing it to the priority list.

ALTERNATE EMBODIMENTS

Several other embodiments are contemplated by the inventors. For example, Click 'n Play is designed to stream any kind of application, not necessarily games. That includes enterprise, business and educational titles etc. The reason Click 'n Play is games oriented is because games are the content that the customers most require and because technologically games are the most demanding applications and therefor require better technology and unique solutions, such as GDTA.

The foregoing description of the embodiments of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. It should be appreciated that many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto.

IT IS CLAIMED THAT

1. A system for enabling rapid launching and execution of applications-on-demand, comprising:
 - i. a streaming server component;
 - ii. a client component;
 - iii. an Installation Bypass Mechanism (IBPM) module; and
 - iv. a Game Data Traffic Analyzer (GDTA) module.
2. The system of claim 1, wherein said streaming server component further comprises:
 - a. a database comprising user information, system information, Installation Bypass Mechanism (IBPM) data and usage statistics;
 - b. a Database Abstraction Layer (DBAL), which is an application program interface (API) that allows applications to access said database without being depended on the database specific database implementation and structure;
 - c. a driver (Virtual File System), for communicating with a system driver on a client side, in order to transfer required data from said database; and
 - d. a Web Server for running a site that is a system front end, enabling said user to log in and execute the application.
3. The system of claim 1, wherein said client component further comprises:
 - a) an Application Level, further comprising:
 - i. a Maximizer for analyzing a user's computer;
 - ii. a Web Interface ActiveX component for running a system driver once said user selected an application to stream; and
 - b) a Driver level, comprising:
 - i. security components for protecting a user's system from illegal use;
 - ii. a Compression Engine for decompressing data streamed from a server to said system; and
 - iii. a File System Driver for sending said data from said system to said application.

4. The system of claim 1, wherein said applications are selected from the group consisting of games, video-on-demand, entertainment programs, edutainment programs, sports programs, education programs and productivity programs.

5. The system of claim 1, wherein said client component is operable on a computing device selected from the group of devices consisting of a PC, Internet TV, notebook computer, personal handheld assistant, smart phone, wearable computer, and mobile computer.

6. A method for enabling rapid launching and execution of streamed applications-on-demand, comprising:

- I. encoding an application, by a GDTA module and an IBPM module; and
- II. enabling rapid launching of said application through bypassing said application's installation, by said IBPM module.

7. A method for enabling rapid launching and execution of applications-on-demand on a client computer, comprising the steps of:

- i. installation tracking, for mapping changes that an original application installation is doing, so that the application can be run without using said original installation; and
- ii. installation bypass, for changing files and settings on the client computer, so that the application-on-demand can be executed without installing said application first using said application's original installation procedure.

8. The method of claim 7, wherein said installation tracking analyzes said original application installation, to map changes said original application installation does during installation, further comprising:

- A. tracking client registry, to compare said registry fields and values of said registry fields before and after application installation, in order to identify registry changes that said application makes during installation;

- B. automatic tracking, for tracking changes that said original application installation is doing to the system;
- C. manual tracking, for detecting installations of additional components and for correcting errors in said automatic tracking process; and
- D. vi. creating at least one CD-image, for emulating an image of an original application CD.

9. The method of claim 7, wherein said installation tracking further comprises the steps of:

- a) installing said application's title on a test computer;
- b) capturing changes that application installation process makes to said computer;
- c) compiling a list of files that were changed and a list of registry keys that were changed on said computer;
- d) analyzing said changes, by an encoder, in order to remove all changes that were logged, but not done by said installation, such that a 'clean' list of registry keys and files that should be modified in order for said application to run properly, result;
- e) adding said list to an installation "script" file, which is used by the client computer; and
- f) performing, by said encoder, an automatic installation process on another machine, in order to reveal any incorrect registry and file changes that might occur due to incorrect analysis.

10. The method of claim 9, further comprising:

- g) if said test computer and said another machine produce identical results, then test process is over; and
- h) if said test computer and said another machine produce non-identical results, said encoder fixes said changes list by modifying any specific field that caused said non-identical results.

11. The method of claim 8, wherein said manual tracking further comprises:

- a. locating minimum system requirements for a specific application, as listed by developer of said application, by an Encoder;
- b. feeding said requirements to an IBPM database;
- c. running a HW/SW diagnostic mechanism to make sure that said application can run on a client computer;
- d. tracking software installations during original installation of said application; and
- e. updating said installations in said IBMP database.

12. The method of claim 7, wherein said installation tracking is executed by a GDTA module, comprising:

- a] tracking of all hard drive accesses under an installation directory made by an operating system, and an application's executable file when said application is running;
- b] outputting a list of accessed files with specified offsets and sizes of blocks; and
- c] gathering information about priority data and said file access list.

13. The method of claim 12, wherein said tracking of all hard drive accesses further comprises:

- i] setting input and output directories names;
- ii] executing an 'Analyze' command;
- iii] initiating a virtual device driver to track all accesses to a specified said input directory;
- iv] filtering said accesses to a certain said directory only;
- v] running an application, by said Encoder, from said directory;
- vi] waiting for a main menu and initiating a special hotkey to write the FAP separator in the ANA-file;
- vii] quitting said application, if said priority data was defined as data that said application requires, until said application gets to said main menu, by said Encoder; and

- viii] alternatively, continuing to use said data, until a point that was defined as such that all said data said application reads in order to get to it is priority data, and then quitting said application.

14. The method of claim 12, wherein said tracking is automatic, and further comprises:

- I. configuring a device driver to hook Windows registry services;
- II. launching said device driver;
- III. launching an application installation application; and
- IV. recording every registry change.

15. The method of claim 7, wherein said installation bypass further comprises:

- I] installing a client program, by downloading a small self-extracting file from a website and following a simple installation process;
- II] selecting an application to launch, by a user client;
- III] checking that the client computer is capable of running said application, by a maximizer;
- IV] authorizing said user and opening a new session;
- V] updating said client's registry and files;
- VI] launching a buffering phase for files that were previously downloaded into said client;
- VII] streaming files from a remote server to said client;
- VIII] sending keep alive messages to said server;
- IX] terminating said application; and
- X] undoing changes to said registry and files.

16. The method of claim 15, further comprising viewing compatibility of a chosen application to a user's system.

17. The method of claim 15, further comprising viewing estimated download time of a chosen application.

18. The method of claim 7, wherein said applications-on-demand are selected from the group consisting of games, kids entertainment and edutainment, sports, education and productivity titles.

19. The method of claim 7, wherein said client computer is selected from the group of computing devices consisting of a PC, Internet TV, notebook computer, personal handheld assistant, smart phone, wearable computer, and mobile computer.

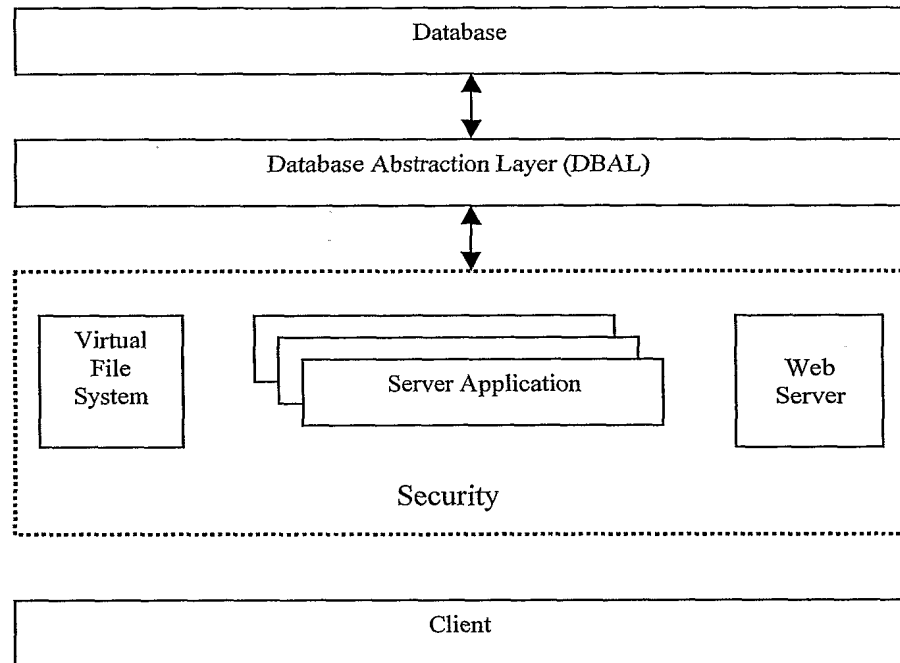
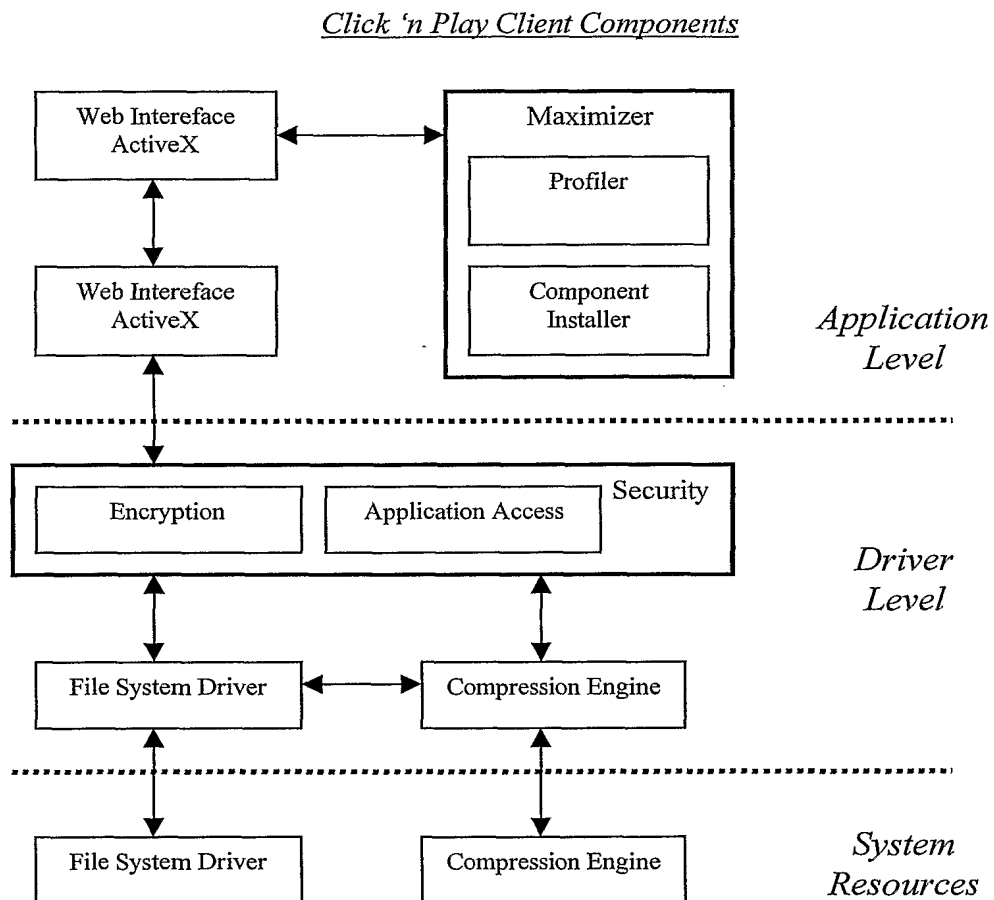
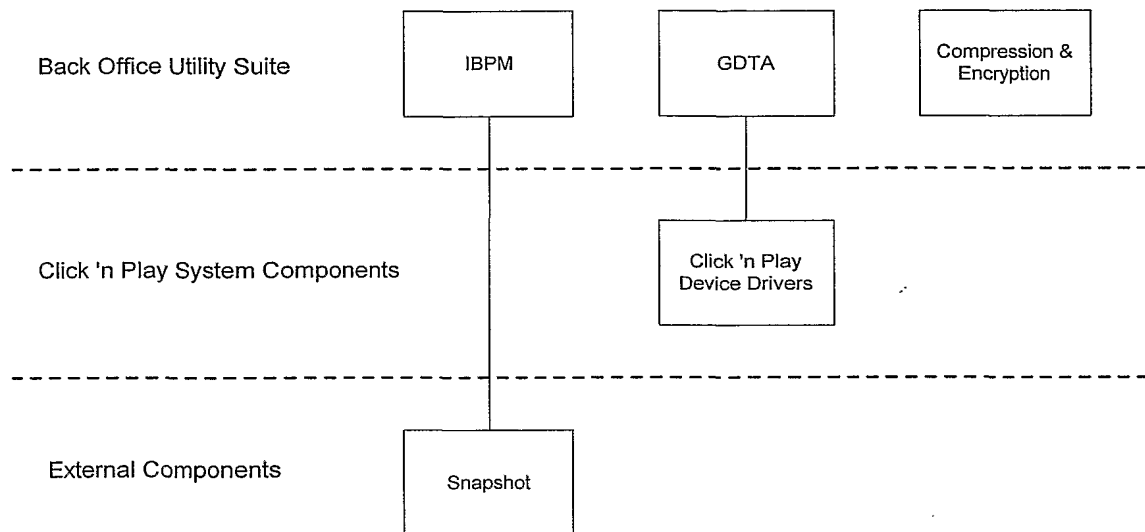
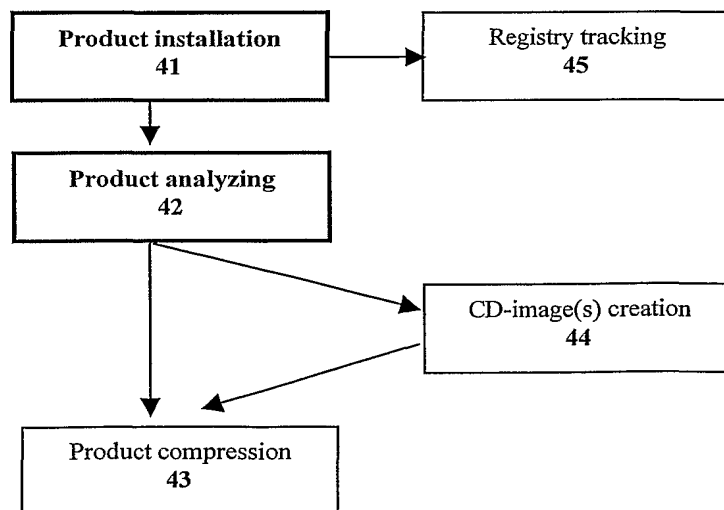
Figure 1*Click 'n Play Server Components*

Figure 2



3/8

Figure 3**Figure 4**

4/8

Figure 5

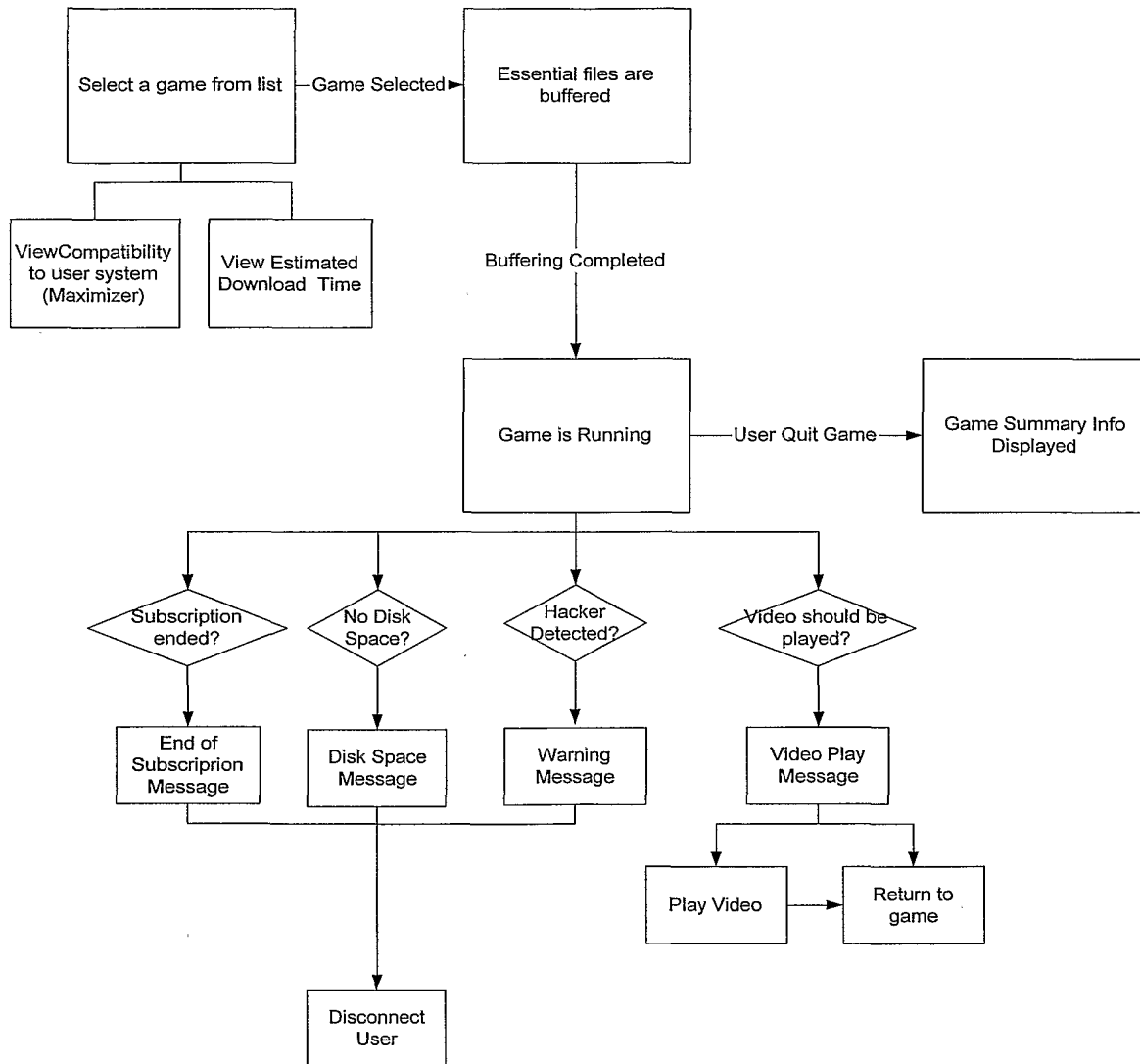
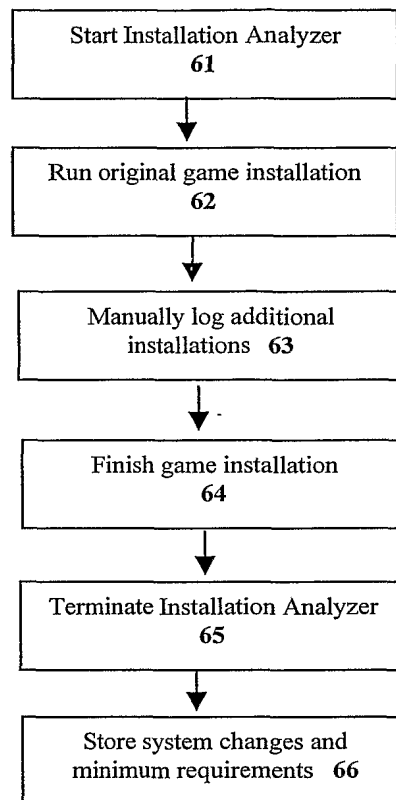


Figure 6

6/8

Figure 7

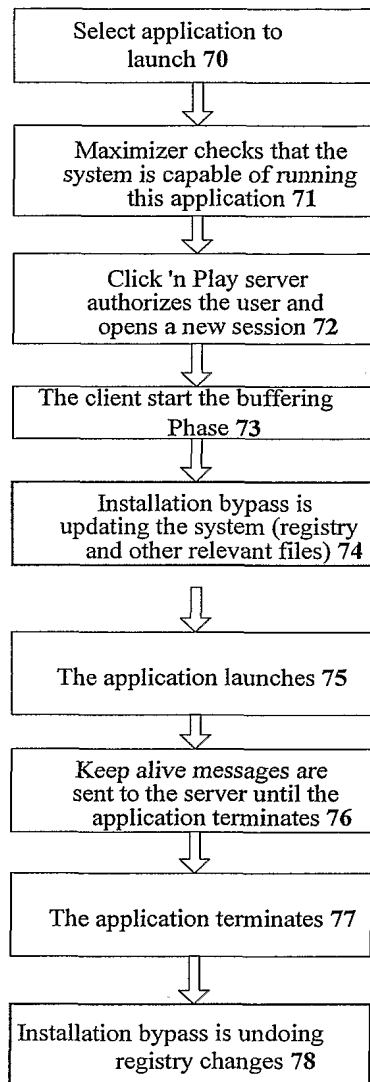


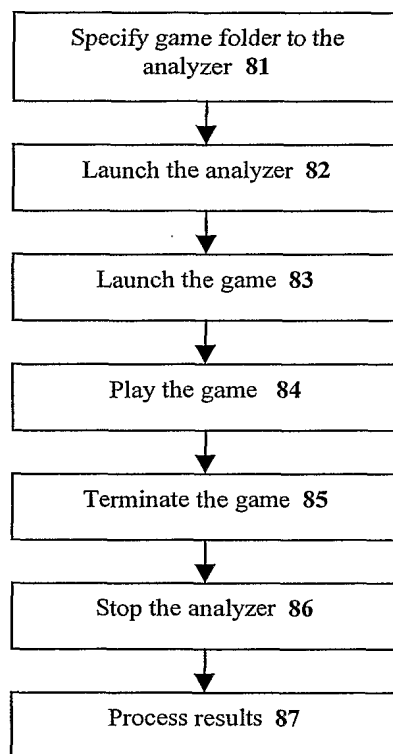
Figure 8

Figure 9

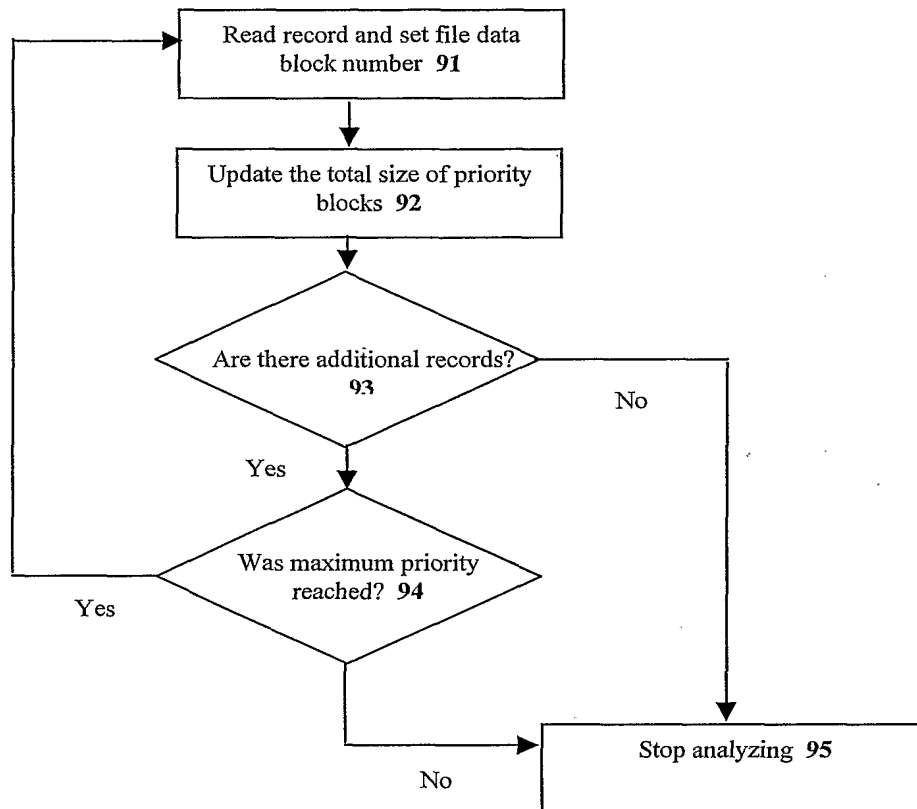


FIGURE 10

Time (ms)	Start offset (bytes)	Length (bytes)	Data Block #
00001	1024	1024	1
00003	4096	16000	2,3,4,5
00005	0	1024	1
00008	24000	1384	6

INTERNATIONAL SEARCH REPORT

Inte: 1 application No.
PCT/US01/20036**A. CLASSIFICATION OF SUBJECT MATTER**IPC(7) :G06F 9/00
US CL :713/1; 717/11

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 713/1, 100; 717/11; 709/231

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A,P	US 6,195,678 B1 (KOMURO) 27 FEBRUARY 2001	1-19
A	US 6,061,504 A (TZELNIC ET AL) 09 MAY 2000	1-19
A	US 5,768,597 A (SIMM) 16 JUNE 1998	1-19
A	US 5,642,417 A (STRINGER) 24 JUNE 1997	1-19



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier document published on or after the international filing date	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

30 AUGUST 2001

Date of mailing of the international search report

01 NOV 2001

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

DENNIS M. BUTLER

Telephone No. (703) 305-3900